

FIGURE 1

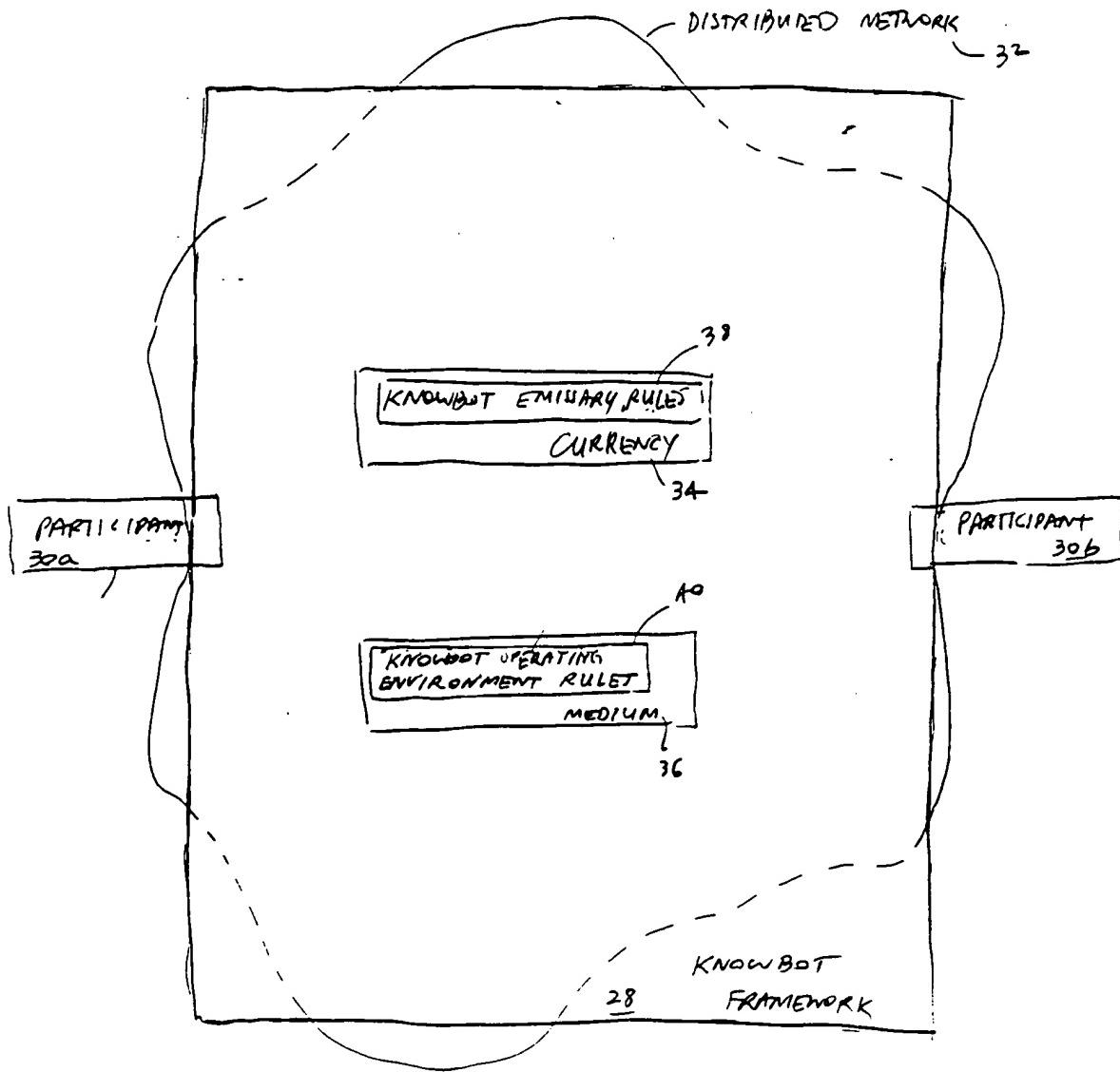


FIGURE 2

962260-250044780

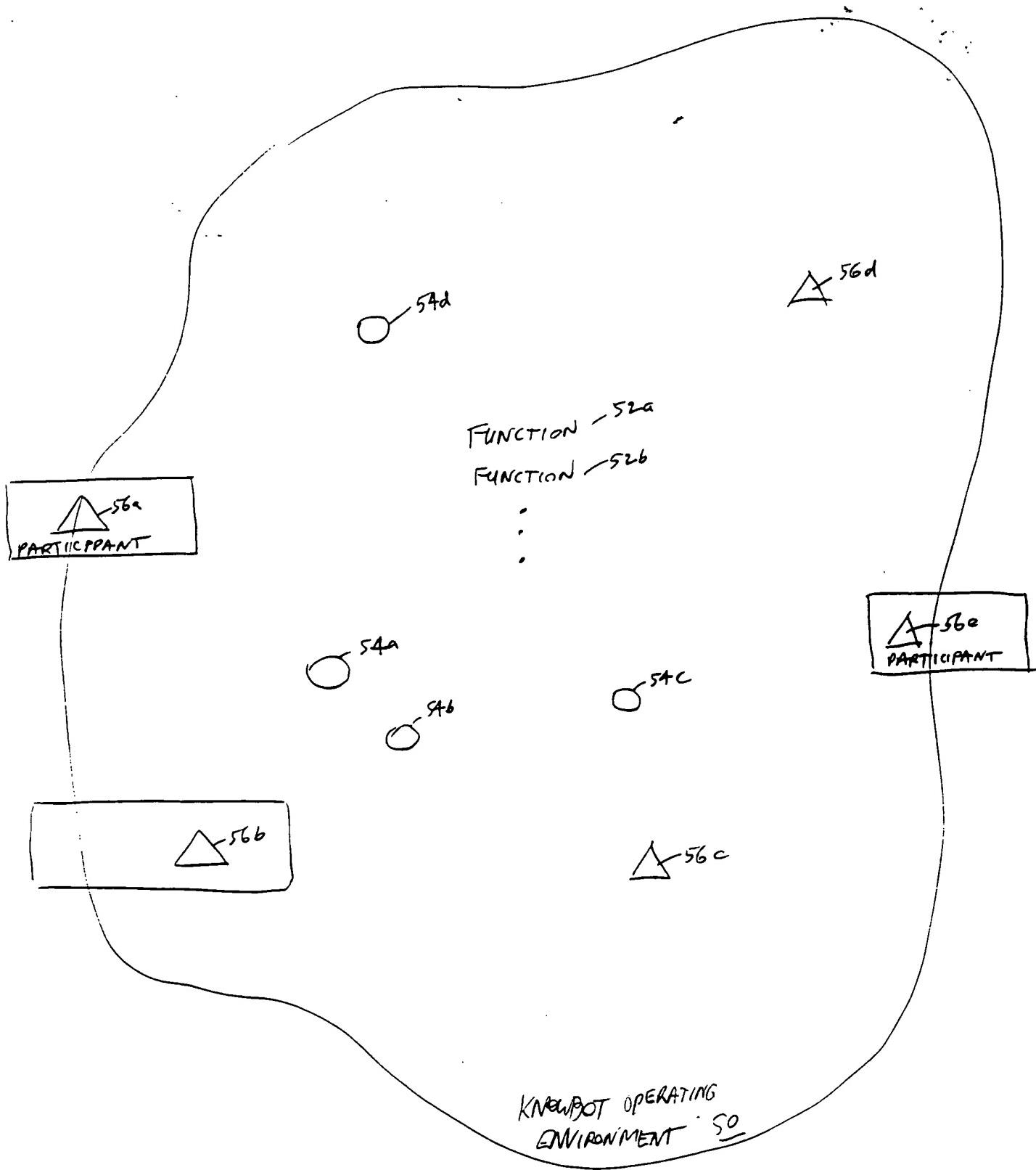
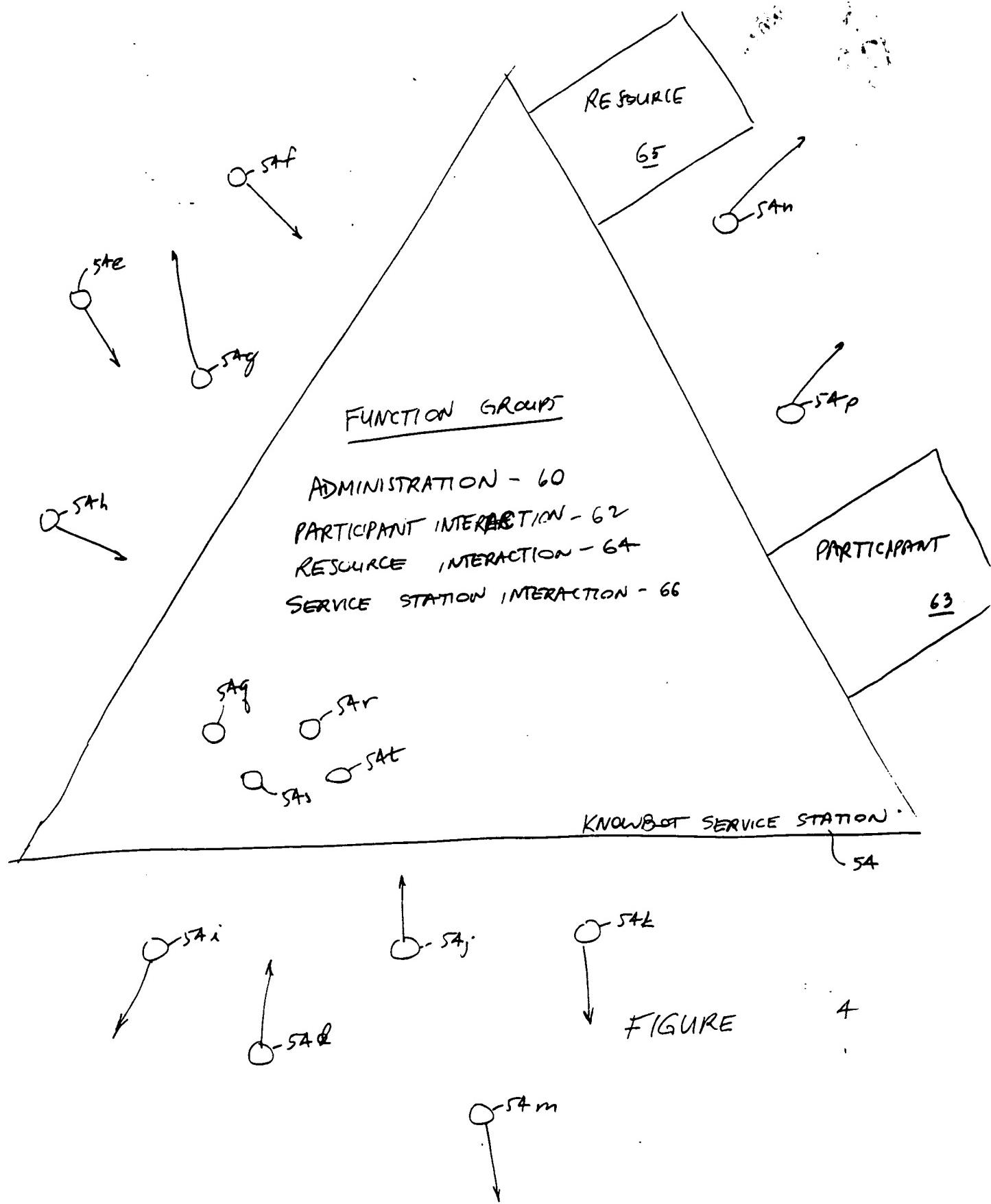


FIGURE 3

03/720092



10/720092

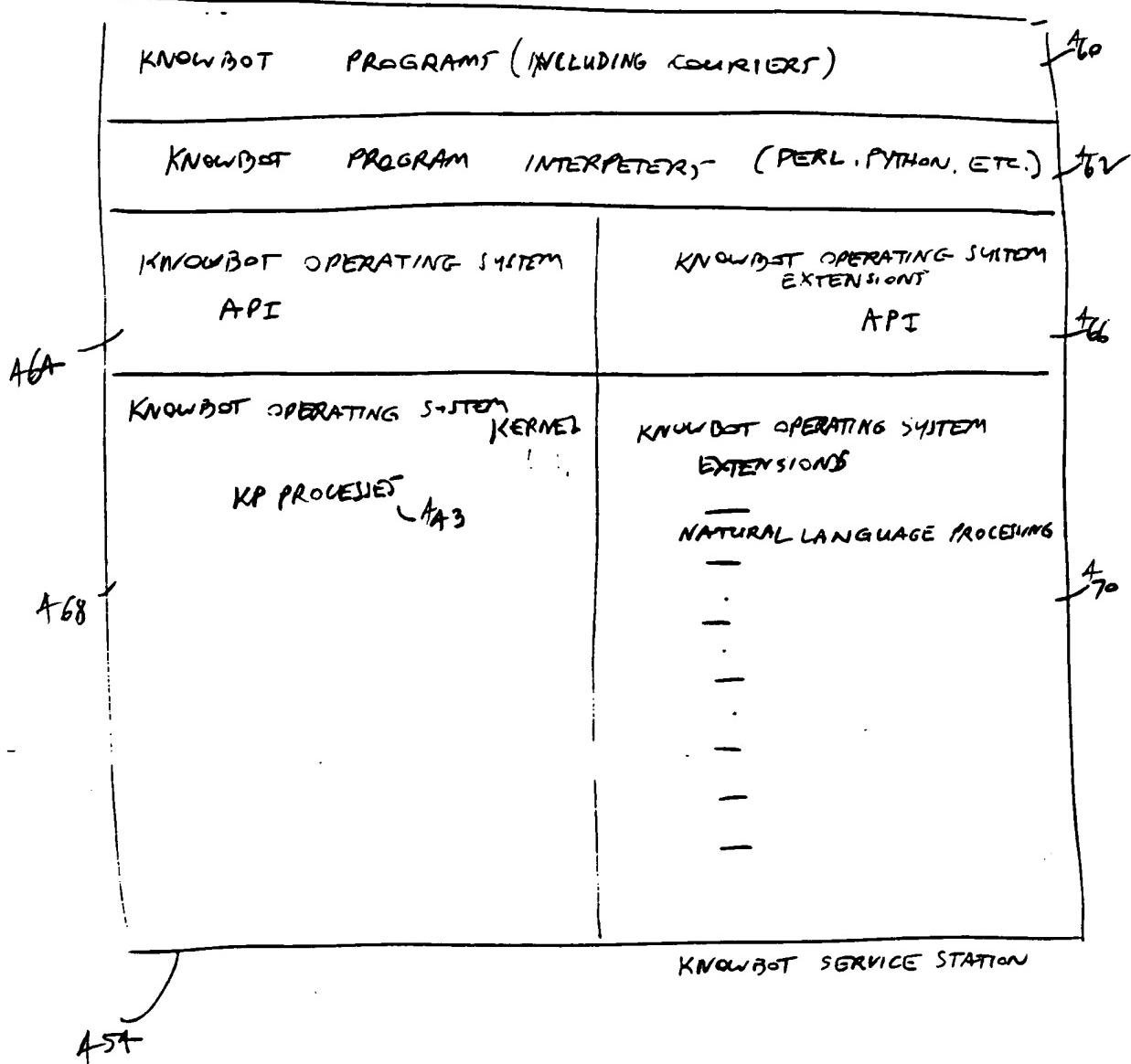


FIGURE : 5

8/720092

ID	AUTH	SCHED	NAV'N	TERMS AND CONDITIONS	
110	112	114	116	118	

SYSTEM TIME DATE	OPR'N	PATH	DESC'N OF DATA	DATA
120	122	124	126	128

FIGURE 6

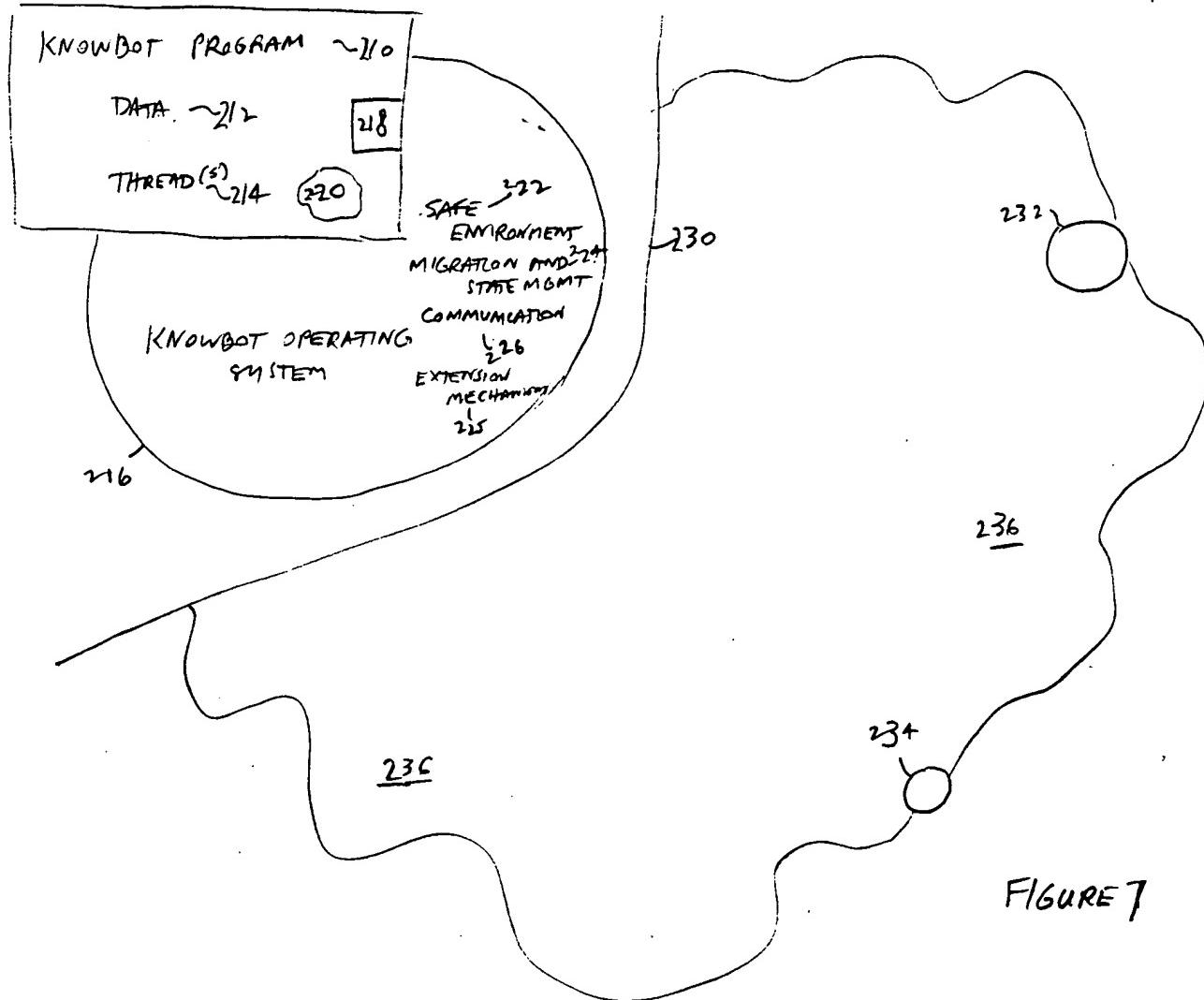


FIGURE 7

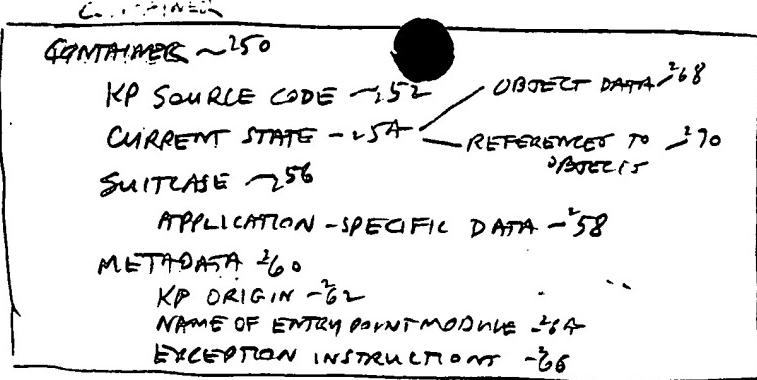


FIGURE 8

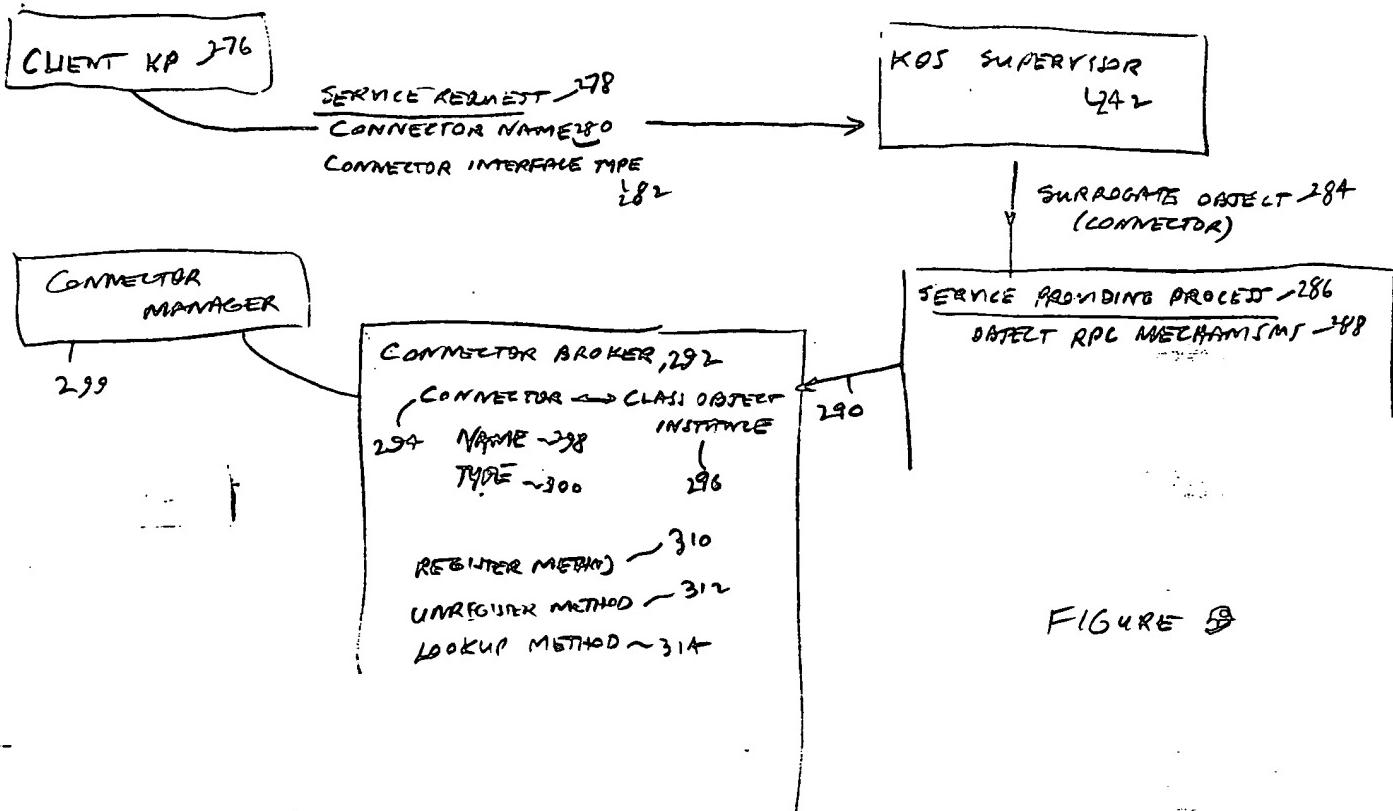


FIGURE 9

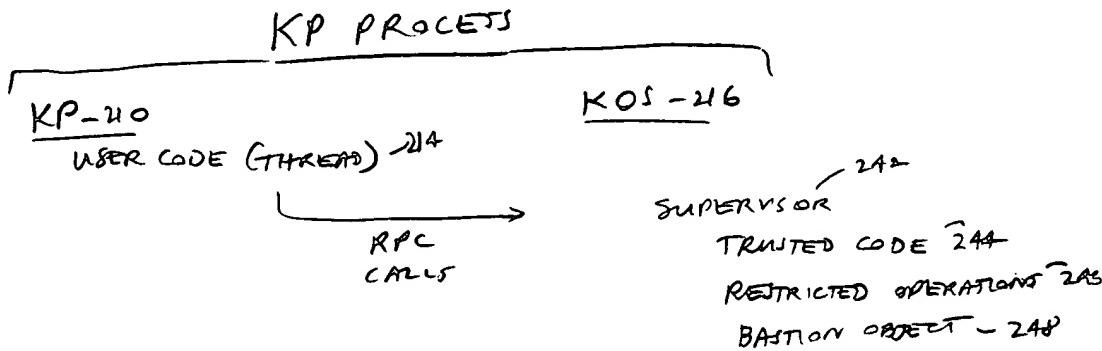


FIGURE 10

03/720092

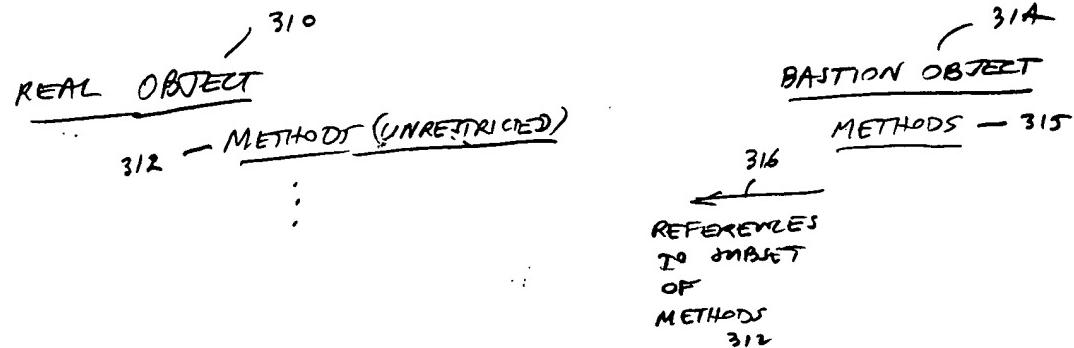


FIGURE 14

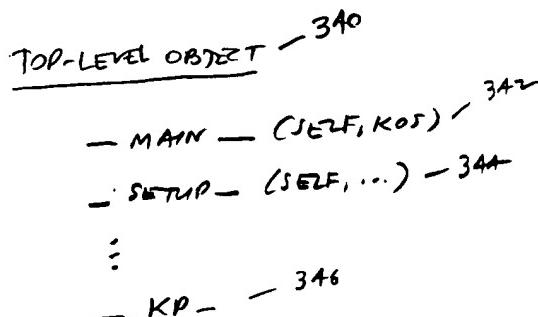


FIGURE 12

```
import rand                                # Python random number module
import nstools                               # helper module for using KOS namespace

class KP:

    def __init__(self):
        "Initialize KP's instance variables."
        self.maxhops = 20
        self.hopcount = 0
        self.visited = []                      # list of KOSes that have been visited

    def __main__(self, kos):
        "Finds services available here, then migrates to a new KOS."
        self.find_services(kos, 'Search.Boolean')
        self.visited.append(kos.get_kos_name())
        self.hopcount = self.hopcount + 1
        if self.hopcount < self.maxhops:
            places = self.get_new_places(kos)
            if places:
                kos.migrate(rand.choice(places))

    def find_services(self, kos, service_type):
        "Save a list of available services in the suitcase"
        services = kos.list_services(service_type)
        file = kos.get_suitcase().open(kos.get_kos_name(), 'w')
        for serv in services:
            file.write(serv.name + '\n')
        file.close()

    def get_new_places(self, kos):
        "Return list of KOSes that have not been visited."
        descriptor = nstools.Lookup(kos.get_namespace(), 'world/kos')
        context = descriptor.Open('Namespace.Context')
        places = []
        for place in context.List():
            if place not in self.visited:
                places.append(place)
        return places
```

-03 / 720092

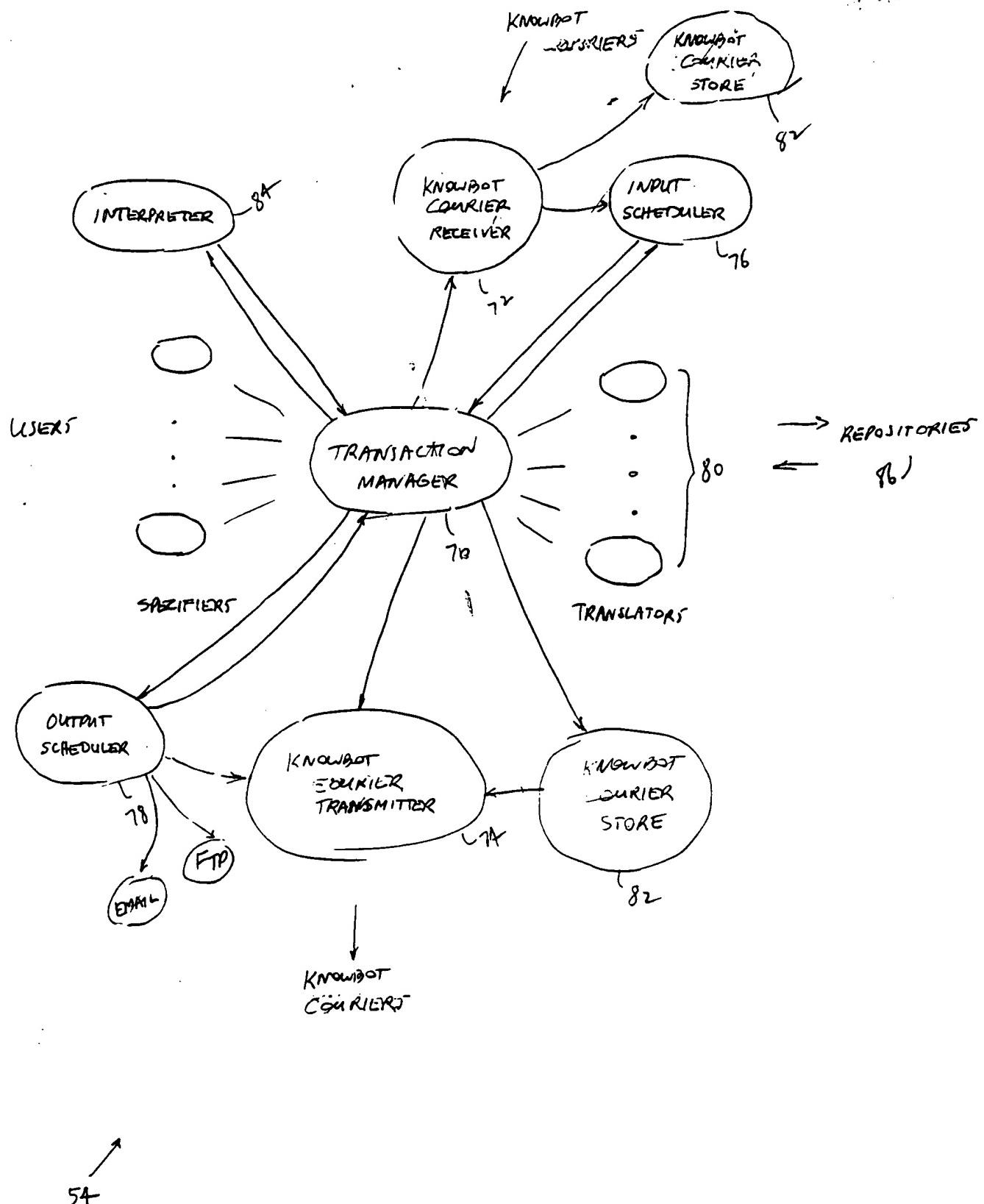


FIGURE 1A